Paying Heed to Collocations

Matthew Stone **Christine Doran** *
Department of Computer Science Department of Linguistics
University of Pennsylvania
Philadelphia, PA 19104
(matthew,cdoran)@linc.cis.upenn.edu

Abstract

In this paper, we introduce a system, Sentence Planning Using Description, which generates collocations within the paradigm of sentence planning. SPUD simultaneously constructs the semantics and syntax of a sentence using a Lexicalized Tree Adjoining Grammar (LTAG). This approach captures naturally and elegantly the interaction between pragmatic and syntactic constraints on descriptions in a sentence, and the inferential and lexical interactions between multiple descriptions in a sentence. At the same time, it exploits linguistically motivated, declarative specifications of the discourse functions of syntactic constructions to make contextually appropriate syntactic choices.

1 Introduction

Words come in a variety of conventional combinations; these units range from short expressions with idiosyncratic meanings, like the *call number* of a book, to full sentences with compositionally-derived, yet frozen, meanings, like *You can't teach an old dog new tricks*. Natural language generation systems must adhere to these combinations, or risk that output will sound as if translated, badly, from Lisp.

Conventional combinations represent not just familiar words, but familiar meanings. Novel descriptions can be unintelligible even if more literally accurate—imagine the *key string* of a book, instead of *call number*. Alternatives to stock language can be even more absurd:[1]

(1) It is futile to attempt to indoctrinate a superannuated canine with innovative maneuvers.

To naturally reuse familiar meanings, generation systems should exploit opportunities to do so as meaning is constructed, not just in transducing meaning to a surface representation. Following this line, the research presented here concerns generating idioms and collocations as part of SENTENCE PLANNING (Kittredge et al., 1991).

Our approach uses Lexicalized Tree Adjoining Grammar (LTAG) and takes DESCRIPTION as the paradigm for the final realization of content. We build on the existing insights of linguists (including (Pustejovsky, 1991; Mel'čuk and Polguère, 1987; Nunberg et al., 1994)) and implementations (including (Reiter and Dale, 1992; Viegas and Bouillon, 1994; Smadja and McKeown, 1991)). However, our proposal introduces two key features. First, the syntax AND SEMANTICS of collocations is planned incrementally and simultaneously. This simplifies the design of the procedure and the linguistic representations it requires; it grounds the decision to select a particular collocation; and it helps integrate the different decisions that must be made in sentence planning. Second, we treat collocations and idioms not just as lexicographic entries, but with full semantics and pragmatics. This allows us to generate specialized uses of words not just in certain lexical or syntactic contexts, but more generally in appropriate discourse contexts. The use of these conventional meanings is a consequence of the systematic design of our planner to observe a computational interpretation of Grice's Maxim of Manner (Grice, 1975): say the usual thing unless you mean something different.

The organization of the paper is as follows. In section 2, we review treatments of collocation in linguistic theory and natural language generation. In section 3 we describe the generation system, SPUD, within which the present analysis will be

*The authors thank Aravind Joshi, Mark Steedman, Martha Palmer, Ellen Prince, Owen Rambow, Mike White, Joseph Rosenzweig, Betty Birner for their helpful comments on various stages of this work. This work has been supported by NSF and IRCS graduate fellowships, NSF grant NSF-STC SBR 8920230, ARPA grant N00014-94 and ARO grant DAAH04-94-G0426.

[1]We found this with some similar examples, at http://149.28.3.6:1701/people/lensky/quotes.html.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1996		**00-00-1996 to 00-00-1996**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Paying Heed to Collocations	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Computer and Information Sciences,University of Pennsylvania,Philadelphia,PA,19104**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**10**	

developed. Then, in section 4, we show how the collocational information can be incorporated into SPUD. Our work is set in the library domain, with the system having the role of a librarian answering patrons' queries.

2 Conventional combinations of words

The different constructions that can be described as collocations exhibit an enormous range of conventionalization. On the one hand are arbitrary, fixed, undecomposable combinations like *by and large*; on the other are locutions like *override a veto* whose preferred co-occurrence derives from the specificity of the semantics of the components. Between these extremes are three classes of constructions of particular concern for natural language generation. First, IDIOMATICALLY COMBINING EXPRESSIONS (Nunberg et al., 1994) must be derived compositionally from special, idiomatic meanings of their parts, as when *strings* = *influence*, *pull* = *exert privately* (from the OED):

(2) The strings she pulled didn't get her the job.

Second, COLLOCATIONS PROPER involve constituents whose meaning is determined by ordinary principles, like *copy area*, but which must be regarded as conventional in light of the oddness of near synonyms (like *duplication zone*); such collocations are the subject of the Lexical Functions of the Meaning-Text Theory (MTT) (Mel'čuk and Polguère, 1987). Finally, SEMANTIC COLLOCATIONS like *long book* derive their particular meaning from the recovery in context of parameters for events and other entities (Pustejovsky, 1991).

Researchers in generation rarely address all of these kinds of conventionality. For example, (Viegas and Bouillon, 1994) handle semantic collocations by implementing Pustejovsky's Generative Lexicon Theory (GLT); modifiers take on specialized meanings derived from salient processes and characteristics associated with the heads they modify. Thus, *a long book* means *a long book to read* because of a lexicographic association between books and reading. Similarly, implementations of MTT describe the conventional use of certain modifiers with heads (Mel'čuk and Polguère, 1987; Iordanskaja et al., 1991; Wanner, 1994) using Lexical Functions. Thus, a function **Magn** determines the realization of a concept very, intense, intensely:

(3) A **Magn** escape ⇒ a narrow escape;
to **Magn** bleed ⇒ to bleed profusely.

Copy area would be handled using the Lexical Function $\mathbf{S_{loc}}$, which returns the name of the location associated with an activity. (Smadja and McKeown, 1991) are an exception in treating a wide range of conventionality, but they simply list the idiomatic status and meaning of a variety of forms in a way that collapses the diistinct theoretical status, and to a large extent, the distinct meanings, of different collocations.

These various existing computational approaches have three main deficiencies. First, they derive conventionality from relational lexicons that describe only the properties of WORDS. However, the features that determine appropriateness of conventional attributions are better modelled as properties of OBJECTS in an evolving model of discourse. Idiomatically combining expressions introduce entities for subsequent reference:

(4) Kim's family pulled some strings on her behalf, but they weren't enough to get her the job. [=(Nunberg et al., 1994) 10c]

Semantic collocations recover their parameters based simply on the things described, regardless of their syntactic proximity, as the examples in (5) show:

(5) a I will not check out a long book.
b I won't check out that book. It's long.
c I won't check that out. It's a long monstrosity.

The modifications achieved by Lexical Functions are parallel: as with *narrow* in (6):

(6) a They made a narrow escape.
b Their escape had been lucky; Bill found it uncomfortably narrow.
c Whew! [after burrowing and swimming out of Alcatraz, amid nearby shots and searchlights] That was narrow!

Second, by treating different conventional combinations as mere paraphrases of one another, researchers complicate the statement of when and why to use conventional forms. No specification of idiomatic combination is complete without representing the pragmatic circumstances in which its use is appropriate (e.g. saying to someone *Your goose is cooked* is not appropriate as a expression of sympathy; the expression conveys a certain amount of disregard for their predicament). Meanwhile, some representation of entities and their salience is required to determine whether ellipsis is possible in context. Whether a *hard idea* is hard to formalize, to communicate, or to understand depends on the topic; to be clear, a natural language system must model how its audience arrives at such understandings.

Third, by recognizing collocations only when transducing underlying semantic representations, researchers limit the extent to which knowledge of collocations can be exploited in generating flu-

ent text. In particular, transduction presupposes that the content of referring expressions has already been established. This means that collocations in definite descriptions either will arise only by accident (or by generate-and-test search) or by a secondary specification that ensures the preference for semantics that can ultimately be realized using collocations.

3 SPUD

This section provides a brief overview of the representations and algorithms that Sentence Planning Using Description (SPUD) uses to address the properties of collocations discussed above. SPUD extends the general procedure for building referring expressions that is suggested by the planning paradigm (Appelt, 1985; Kronfeld, 1986). The procedure starts from a set of entities to describe and a set of intentions to achieve in describing them. It then applies operators that enrich the content of the description until all intentions are satisfied. As in realizations like (Dale and Haddock, 1991), we constrain the inference required to generate and evaluate alternatives by limiting the kinds of intentions considered. However, whereas the planning procedures on which we base our system are used only for noun phrases, we apply this procedure to the sentence as a whole using a rich semantic representation; further, although these procedures typically construct an abstract semantic representation, we treat operators as entries with syntactic, semantic and pragmatic properties. The lexicalized tree adjoining grammar (LTAG) formalism provides an abstraction of the combinatorial properties of words. The resulting system offers a number of advantages. By incorporating content into descriptions of a variety of entities until the addressee can fill in the details, this procedure results in short, natural and unambiguous sentences. Moreover, by evaluating and selecting alternatives on the basis of their pragmatic, semantic and syntactic contribution to the sentence as a whole, the procedure uniformly handles a variety of interactions inside a sentence, including collocations.

3.1 Linguistic Specifications

This algorithm requires a declarative specification of three kinds of information: first, what operators are available and how they may combine; second, how operators specify the content of a description; and third, how operators achieve pragmatic effects. We represent operators as elementary trees in an LTAG, and use TAG operations to combine them; we give the meaning of each tree as a formula in an ontologically promiscuous representation language; and, we model the pragmatics of operators by associating with each tree a set of discourse constraints describing when that operator can and should be used.

TAG (Joshi et al., 1975) is a grammar formalism built around two operations that combine pairs of trees: SUBSTITUTION and ADJOINING. A TAG grammar consists of a finite set of ELEMENTARY trees, which can be combined by these operations to produce derived trees recognized by the grammar. In substitution, the root of the first tree is identified with a leaf of the second tree, called the substitution site ($\downarrow$). Adjoining is a more complicated splicing operation, where the first tree replaces the subtree of the second tree rooted at a node called the adjunction site; that subtree is then substituted back into the first tree at a distinguished leaf called the FOOT node ($*$). Elementary trees without foot nodes are called INITIAL trees and can only substitute; trees with foot nodes are called AUXILIARY trees, and must adjoin. TAG elementary trees abstract the combinatorial properties of words in a linguistically appealing way. Figure 1(a) shows an initial tree representing *the book*. Figure 1(b) shows an auxiliary tree representing the modifier *syntax*, which could adjoin into the tree for *the book* to give *the syntax book*. All predicate-argument structures are localized within a single elementary tree, even in long-distance relationships. Figure 1(c) shows the topicalized tree anchored by *have*; both of its arguments are substitution sites.

Our grammar incorporates two additional principles. First, the grammar is LEXICALIZED (Schabes, 1990): each elementary structure in the grammar contains at least one lexical item. Second, our trees include FEATURES, following (Vijay-Shanker, 1987).

We specify the semantics of trees by adapting two principles of computational semantics to the LTAG formalism. First, as originally advocated by Hobbs (1985), we adopt an ONTOLOGICALLY PROMISCUOUS representation that includes a wide variety of types of entities. In particular, abstract entities are introduced to represent the SCOPES of OPERATORS. A predicate is interpreted as if inside a scope when the predicate takes the corresponding abstract entity as an argument. For this paper, we need EVENTUALITIES as abstract representations of spatiotemporal scope and INFORMATION STATES to abstract the scope of modal operators like possibility and belief. Nodes are labeled as supplying information **about** a particular entity or

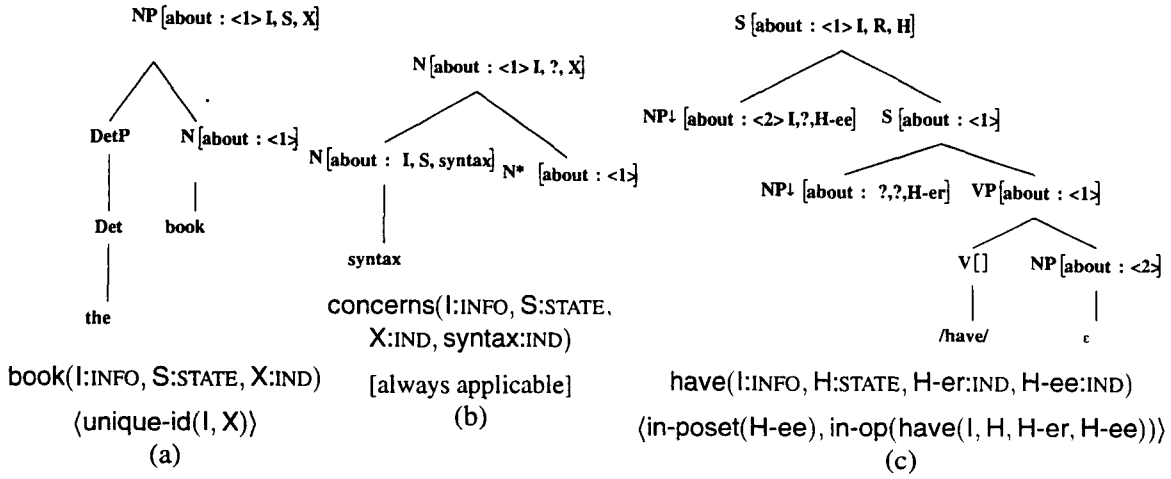

Figure 1: LTAG trees with semantic and pragmatic specifications

collection of entities (this is inspired by a similar hypothesis in (Jackendoff, 1990)). To guarantee a coherent meaning for a derived structure, a node **about x** can only substitute or adjoin into another node **about x**. Here, we simply use an additional feature on the node to capture this. Figure 1 also shows the semantics and **about** labels for each tree; **?** indicates unspecified **about** values.

To package information appropriately requires sensitivity to the knowledge of the hearer and the state of the discourse. Different constructions make different assumptions about the status of entities and propositions. We model these differences by including in each tree a specification of the contextual conditions under which use of the tree is pragmatically licensed. Our conditions derive from linguistic analysis, particularly (Gundel et al., 1993; Ward, 1985; Ward and Prince, 1991; Prince, 1993; Birner, 1992).

The status of entities and propositions in discourse varies along at least four dimensions that are relevant to these specifications. First, entities differ in NEWNESS (Prince, 1981). At any point, an entity is either new or old to the HEARER, according to whether or not the hearer has at least implicit knowledge of the existence of the entity. Analogously, an entity is either new or old to the DISCOURSE, according to whether the discourse contains an earlier reference to it. Second, entities differ in SALIENCE (Grosz and Sidner, 1986; Grosz et al., 1995). At any point, salience assigns each entity a position in a partial order that indicates how accessible it is for reference in the current context. Third, entities are related by material PARTIALLY-ORDERED SET (POSET) RELATIONS to other entities in the context (Hirschberg, 1985). These relations include part and whole, subset and superset, and membership in a common class; a number of constructions depend on poset relations to signal their connection with context. Finally, the discourse may distinguish some OPEN PROPOSITIONS, propositions containing free variables, as being under discussion (Halliday, 1967; Prince, 1986). This privileges subsequent information that provides true instantiations for the variables in a salient open proposition. We assume that information of these four kinds is available in a model of the current discourse state, and that the applicability conditions of constructions can freely make reference to this information. The pragmatic specification for *the book*, *syntax*, and topicalized *have* appear under the semantics for each tree in figure 1.

Our discourse model contains information on the shared knowledge of the speaker and hearer, private knowledge of the speaker, and a specification of entities and their discourse status. In the library domain, shared knowledge includes such things as rules about how to check out books, while speaker knowledge includes such information as the status of books in the library. The discourse model can also include general properties that describe the conversational situation as a whole; for example, it might specify the formality of the register in which the communication is being conducted.

3.2 The algorithm

Our system takes two types of goals. First, goals of the form *identify x as cat* instruct the algorithm to construct a description of entity *x* using the syntactic category *cat*. If *x* is uniquely identifiable, then this goal is only satisfied when the overall content planned so far distinguishes *x* for the hearer. If *x* is hearer new, this goal is satisfied by including any constituent of type *cat*. Sec-

ond, goals of the form *communicate p* instruct the algorithm to include the proposition *p*. This goal is satisfied as long as the overall content ENTAILS *p* given the shared knowledge of speaker and hearer.

In each iteration, our algorithm must determine the appropriate elementary tree to incorporate into the current description. It performs this task in two steps to take advantage of the regular associations between semantics and trees in the lexicon. Lexical entries pair a semantic constraint with a FAMILY of TREES that describe the combinatory possibilities for realizing the semantics. For example, *book* is stored with a tree family that includes *a book* and *the book*. We have chosen to include the determiners in the basic NP trees because of their importance for the semantics and pragmatics of the NP. Similarly, there are different initial trees for each clause type anchored by a particular verb. Trees in the tree family are shared among all lexical items that share a particular structure. This allows us to specify the pragmatic constraints associated with the tree type once and for all, regardless of which verb selects it. Moreover, we can determine which tree to use by looking at each tree ONCE, even when the same tree is associated with multiple lexical items.

Hence, the first step is to identify applicable lexical entries: these items must correctly describe some entity; they must anchor trees that can substitute or adjoin into a node that describes the entity; and they must contribute toward satisfying current goals. (We describe more precisely how this contribution is evaluated in section 4.1.) Then, the second step identifies which of the associated trees are applicable, by testing their pragmatic conditions against the current representation of discourse. We combine possible lexical items and possible trees, to give an evaluation of all applicable options. The algorithm identifies the entries that most contribute to current goals, and from these, selects the entry with the most specific semantic and pragmatic licensing conditions. This means that the algorithm generates the most marked licensed form for the particular context.

The entry is then substituted or adjoined into the tree at the appropriate node. The meaning of the derived tree is simply the CONJUNCTION of the meanings of the elementary trees used to derive it. The entry may specify additional goals, because it describes one entity in terms of a new one. These new goals are added to the current goals, and then the algorithm repeats.

3.3 Discussion

The strength of the present work is that it captures a number of phenomena discussed elsewhere separately, and does so within the unified framework of description. In particular, we treat many types of content as contributing to expressions that refer to semantic objects. The tenses of sentences in discourse refer to times in much the same way pronouns and full NPs refer to individuals (Partee, 1973; Partee, 1984). The modality of sentences may refer to a salient possibility (Roberts, 1986) or provide the content of a salient psychological state (Wiebe, 1994). The rhetorical connection between a sentence and surrounding discourse should also be described with adjuncts (Huang, 1994). Adjuncts giving details about an event should be included only after reasoning that these adjuncts are in fact necessary in context (McDonald, 1992).

With its incremental choices and its emphasis on the consequences of functional choices in the grammar, our algorithm resembles the networks of systemic grammar (Mathiessen, 1983; Yang et al., 1991). However, unlike systemic networks, our system derives its functional choices dynamically using a simple declarative specification of function that correlates well with recent linguistic work. Further, like many sentence planners, we assume that there is a flexible association between the content input to a sentence planner and the meaning that comes out. Other researchers (Nicolov et al., 1995; Rubinoff, 1992) have assumed that this flexibility comes from a mismatch between input content and grammatical options. In our system, such differences arise from the referential requirements and inferential opportunities that are encountered.

Previous authors (McDonald and Pustejovsky, 1985; Joshi, 1987) have noted that TAG has many advantages for generation as a syntactic formalism, because of its localization of argument structure. These aspects of TAGs are crucial for us. Lexicalization allows us to easily specify local semantic and pragmatic constraints imposed by the lexical item in a particular syntactic frame. Various efforts at using TAG for generation (McDonald and Pustejovsky, 1985; Joshi, 1987; Yang et al., 1991; Nicolov et al., 1995; Wahlster et al., 1991) enjoy many of these advantages. Furthermore, (Shieber et al., 1990; Shieber and Schabes, 1991; Prevost and Steedman, 1993; Hoffman, 1994) exploit similar benefits of lexicalization and localization. What sets SPUD apart is its simultaneous construction of syntax and semantics, and the tripartite, lexicalized, declarative gram-

matical specifications for constructions it uses. (Shieber et al., 1990; Shieber and Schabes, 1991) construct a simultaneous *derivation* of syntax and semantics—but they do not *construct the semantics*: it is an input to their system. Moreover, they do not represent any pragmatic information. (Prevost and Steedman, 1993; Hoffman, 1994) do represent the division of sentences into theme and rheme, but because they do not model the pragmatics of particular constructions, they plan descriptions in a separate step.

4 Conventional combination in SPUD

Because LTAG can associate multiple lexical items to a single tree, it is straightforward to list frozen idioms, like *call number*, in the lexicon (Abeille and Schabes, 1989). These specifications can include idiosyncratic semantic and pragmatic information; grammatical processes like tense marking apply normally.

In this section, we describe how SPUD can be made to use words in other conventional combinations. Our proposal involves three steps. First, as in (Reiter and Dale, 1992), we stipulate that some attributes of entities are more important than others, and that some words more naturally describe those attributes. Second, in keeping with ontological promiscuity (Hobbs, 1985), we represent the importance of attributes by the salience of events and states in the discourse model—these states and events now have the same status in the discourse model as any other entities. Finally, we extend SPUD's evaluation of alternatives, so that it describes the most salient entities possible, and uses basic-level terms wherever possible. By associating entities not just with salient attributes but also with salient actions and salient figurations, we capture collocations, semantic collocations and idiomatic compositionality using a uniform mechanism.

4.1 Collocations proper

Although primarily concerned with the interpretation of Gricean maxims, the work of (Reiter and Dale, 1992; Dale and Reiter, 1995) underlines the conventionality of description. Based on a review of psychological experimentation and their own study of referring expressions in task-oriented dialogue, they argue that some referring expressions can be constructed simply by selecting properties from a prioritized list of attributes until the entity is distinguished. To further conventionalize descriptions, they privilege the selection of properties that provide basic-level characterizations of the entity (Rosch, 1978; Reiter, 1991). Because any property is considered for only one attribute, this algorithm offers a linear speedup over the greedy strategy used in (Dale and Haddock, 1991) and described above for SPUD, which considers every property at every stage. However, here we focus on how incorporating similar ideas into SPUD gives a general framework for specifying conventional uses of words, and remain neutral about achieving similar speedups.

Reiter and Dale suggest that the prioritized list of attributes their algorithm uses is domain-dependent. In fact, we find that these lists are both domain and object-dependent. Obviously the attributes by which we describe abstractions like events and states—typically time, location, and manner or quality—are quite distinct from the natural attributes by which physical objects are distinguished. However, in the library, widely different attributes can be appropriate even for physical objects of various types. Books can be described by author, by physical characteristics, or by content (e.g. *Chomsky's book, the yellow book, a math book*). Periodicals, meanwhile, are best described by date of issue (e.g. *the May issue of Language*). Parts of the library, as we shall see below, are best distinguished by the special services they provide (e.g. *the reference desk*).

SPUD's ontologically promiscuous discourse model offers a natural dimension to represent these distinctions. Since each property of an object is associated with an eventuality argument, we can assign a level of salience for that eventuality. We can use this ranking to indicate the conventional importance of the eventuality in distinguishing the object. In other words, if we know $p(e, x)$, and it is natural to describe x in terms of p, e will be salient. For example, since periodicals are easily identified by their date of issue, we should make this state salient. Note then that salience is determined for explicitly mentioned and inferable entities and depends not only on recency of mention but also on facts about the conversational situation and real-world relationships between objects.

Reiter and Dale also point out that which characterizations are basic-level must be adjusted to reflect the expertise of the addressee; however, we shall sidestep this issue here by assuming that certain lexical items are simply listed as basic-level terms.

By itself, these additions are not enough: SPUD must also take salience and basic-level semantics into account in the evaluation of its alternatives. That is: other things being equal, SPUD should choose to incorporate at each stage the

syntactic-semantic-pragmatic unit which refers to maximally salient entities; and, other things being equal, SPUD should incorporate a basic-level predicate. Integrating Reiter and Dale's prioritization of these considerations with SPUD's other considerations leads to the following ranking of criteria for comparison:

(7) RULES OUT A DISTRACTOR OR ENTAILS NEEDED INFORMATION > SALIENCE OF ENTITIES MENTIONED > NUMBER OF DISTRACTORS RULED OUT > NUMBER OF INFORMATIONAL GOALS ACHIEVED > BASIC-LEVEL TERM > SPECIFICITY OF LICENSING CONDITIONS

With the right linguistic specification, this is all the machinery SPUD needs to generate conventionalized forms. To see how we can generate ordinary collocations, consider describing parts of a library. Descriptions of these places are typically collocations: e.g. *copy area*, *reference desk*, *interlibrary loan office*. The names can be abbreviated in context, they can be interpreted compositionally, but substituting synonyms generally sounds odd. Nevertheless, these descriptions share features, in that one always describes its type, sometimes the service it provides, and most rarely its location. This leads to the following axiomatization of the salience of states:

(8) part-of(I, S1, Part, Lib) ∧
library(I, S2, Lib) ⊃
(has-type(I, S3, Part, Type) ∧
provides-service(I, S4, Part, Service) ∧
has-location(I, S5, Part, Loc) ⊃
S3 $>_S$ S4 $>_S$ S5)

The first argument of each predicate is the information state in which the various predications hold; the second argument is the eventuality which witnesses the application of the predicate; $>_S$ indicates the salience ranking of the states. Thus, (8) considers a case where there is a part Part of a library Lib: suppose S3 witnesses that Part has some type Type; S4, that Part provides service Service; and S5, that Part has location Loc. Then, S4 is more salient than S5, and S3 is more salient than both. We must specify not only the salience of different states for the same copier, but also the salience of corresponding states for different copiers. Another axiom, similar to (8), ensures that states that specify a given attribute are equally salient across copiers when the copiers involved are equally salient.

The vocabulary chosen, meanwhile, reflects conventional names for the structures and services of the library. Semantic declarations such as the following represent this:

(9) *area* (I, S, A) : BASIC
has-type(I, S, A, area)

That is, *area* uses the specified semantics to provide a basic-level description of A in terms of state S and information I. Note that SPUD always chooses a maximally specific licensed form out of equally good alternatives. Thus, we can have any number of basic-level terms to describe an object, and the appropriate one will be selected on the basis of its specificity. For example, even if both *room* and *area* are basic, a room will be still be described using *room*, because all rooms are areas but not all areas are rooms.

Together, these assumptions suffice to generate collocations for library parts. For example, suppose SPUD has the goal of describing the part of the library where copying takes place, location e30. SPUD first selects the NP *the area*, eliminating alternatives like *the room*, *the desk*, *the stack*, because they do not truthfully describe e30. However, since many other parts of the library are also *areas*, the current description does not rule out all possible distractors, and SPUD further elaborates the description. The modifiers *copy* and *service* are both applicable to e30, but *copy* eliminates all distractors while *service* does not, so the former is selected, yielding the final NP *the copy area*.

4.2 Semantic collocations

To handle semantic collocations now requires only a representation of how certain lexical items depend on hidden parameters for actions and events. For example, consider the lexical item *fast*: it constrains the typical rate of some action performed by or with the entity it describes. Thus, it has a meaning like this:

(10) *fast* (I, S, Obj, Act) : BASIC
participant(I, S2, Obj, Act) ∧
typical-rate(I, S3, Act, Rate) ∧
high(I, S, Rate)

Corresponding to the qualia structure of GLT, we have axioms describing what actions are associated with objects and how salient they are. For a photocopier, this might be specified this way:

(11) photocopier(I, S, X) ⊃
(participant(I, S1(X), X, copy-action) ∧
participant(I, S2(X), X, repair-action) ∧
participant(I, S3(X), X, fill-paper-action) ∧
S1(X) $>_S$ S2(X) $>_S$ S3(X))

That is, typically, with copiers, you not only make copies, but also fill them with paper, and (sadly, all too often), have them repaired; however, copying is the most salient thing to do with them. Note that while this axiom is expressed at the same level of

generality as GLT's qualia structures, this rule is part of world knowledge and applies to all things that are photocopiers, not to all occasions where things are described as photocopiers.

To see how SPUD uses these specifications, let us say that we have a copier, c42, which is the sole fast copier (at making copies) in the library. After planning a referring expression *the copier*, SPUD has the goal of distinguishing c42 from the other copiers. The KB entails the fact fast(i,s,c42,copy-action), which allows us to incorporate the lexical item *fast* into the description. SPUD then evaluates the distractor set; since copy-action is a new reference, SPUD checks whether any distractor is also fast at an action which is at least as salient as copy-action. None are, because copy-action is the most salient action of copiers. Since the expression, *the fast copier*, now refers uniquely both to c42 and to copy-action, the referring expression is adequate. The need to rule out distractor actions can cause information to be added to an expression. To describe another copier, c43, which is the fastest copier to fill with paper, SPUD would describe not only its rate but also the relevant action in order to distinguish it from c42, i.e. the fast copier to fill. Also, note SPUD can use this same meaning of *fast* and the same reasoning process even when *fast* does not modify a noun. (For example, in a slightly different context it could describe the state s with this sentence: *The copier is fast.*)

4.3 Idiomatic composition

As (Nunberg et al., 1994) emphasize, idiomatic composition typically involves some distinctive figurative or metaphorical view of the objects being described. Accordingly, to specify idiomatic composition, we adopt a representation of such views from (Ballim et al., 1991). They outline a model of reasoning in which facts are partitioned into sets called ENVIRONMENTS. Environments can collect information about particular topics, or, when nested, can represent the beliefs of particular agents. Moreover, they suggest that nonliteral language can also be represented using a nested environment, whose contents are determined by treating topic-environments as competing sources of information analogous to different agents' views. We believe reasoning algorithms like those presented in (Ballim et al., 1991) should be an important part of any natural language generation system which aims at idiomatic language; however, for the present, the key feature of this account is just its principled use of multiple information-states, in which different facts hold.

We combine this representation with two assumptions about how information states are represented in the grammar. We assume that information states are recovered from the context just like other parameters of interpretation like states and actions. However, we use trees that in some cases impose coreference requirements between the information states in which different constituents are interpreted. For the examples we have considered, what seems right is to coindex the information states of modifiers and their heads, and to coindex the information state of a verb with all its arguments except the subject. (The trees of figure 1 respect this generalization.)

Consider the example from section 2: the combined convention *strings = influence, pull = exert privately*. The opportunity to use the expression arises in any information state k where:

(12) $\mathsf{influence}(k, S1, C, X, F) \wedge$
$\mathsf{subverts}(k, S2, C, \mathsf{bureaucracy}) \wedge$
$\mathsf{exert}(k, E, X, C) \wedge \mathsf{private}(k, S3, E)$

We can represent the idiom semantically using a rule that introduces the associated stock figuration, that bureaucrats are puppets whose behavior is governed by such influence: bp(k, C).

(13) $\mathsf{strings}(\mathsf{bp}(k, C), S4(k, C), C) \wedge$
$\mathsf{pull}(\mathsf{bp}(k, C), E, X, C)$

Now we just use the ordinary meanings of *pull* and *strings* to describe this situation.

To constrain the situations in which this is an appropriate thing to say, we need to determine the circumstances in which bp(k, C) is as salient as k. (One might claim that the ready salience of the information state—naturally, different across languages—is what makes idioms different from metaphors.) Although such a specification is clearly open-ended, we approximate the full set of constraints in terms of two parameters of the discourse context: a reasonable degree of intimacy between speaker and hearer and an informal register of conversation.

Consider how the noun phrase *the strings she pulled* is generated to describe some exerted influence c. Under appropriate discourse conditions, SPUD can choose to describe c in terms of the information state bp(k, c) and the lexical item *strings*. To rule out c's additional distractors, the object relative clause anchored by *pulled* is chosen; the informational coindexation between the foot N node and the verb in an object relative clause ensures that *exerted* does not apply—because c is NOT the object of an exerting event according to information bp(k, c). Finally, the

agent of the pulling is described with *she*.

5 Conclusion

SPUD uses a single body of syntactic, semantic, and pragmatic knowledge to generate both productive and conventional descriptive expressions. Hence, SPUD offers a natural framework for dealing with the interactions between syntax, semantics and pragmatics which characterize the sentence planning problem, and ensuring contextually appropriate output. This knowledge produces good results; however, it is very expensive to build. The system requires rich descriptions of language and of the world, which for now must be specified by hand. Only SPUD's underlying reasoning mechanisms are completely application independent, but others are at least partly reusable. Specifications of world knowledge can be used for generation in many languages, while linguistic specifications apply across many domains. For different languages, SPUD's model may vary along a number of dimensions, including the exact range of objects which roughly corresponding lexical items can describe, and the (default) salience rankings—both for typical properties and actions associated with objects and for the information states licensing idioms. Such differences will allow SPUD to generate different collocations in different languages, even when describing the same entities.

We have implemented a preliminary version of SPUD, and realized the examples discussed in section 4. Our future work includes refining this implementation and enriching its linguistic knowledge.

References

Anne Abeille and Yves Schabes. 1989. Parsing Idioms in Lexicalized TAGs. In *Proceedings of EACL '89*, pages 161–65.

Douglas Appelt. 1985. *Planning English Sentences*. Cambridge University Press, Cambridge England.

Afzal Ballim, Yorick Wilks, and John Barnden. 1991. Belief ascription, metaphor, and intensional identification. *Cognitive Science*, 15:133–171.

Betty Birner. 1992. *The Discourse Function of Inversion in English*. Ph.D. thesis, Northwestern University.

Robert Dale and Nicholas Haddock. 1991. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.

H. P. Grice. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics III: Speech Acts*, pages 41–58. Academic Press, New York.

Barbara Grosz and Candace Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204.

Barbara Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering: A frameword for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307.

M. A. K. Halliday. 1967. Notes on transitivity and theme in English. *Journal of Linguistics*, 3:117–274.

Julia Hirschberg. 1985. *A Theory of Scalar Implicature*. Ph.D. thesis, University of Pennsylvania.

Jerry R. Hobbs. 1985. Ontological promiscuity. In *Proceedings of ACL*, pages 61–69.

Beryl Hoffman. 1994. Generating context-appropriate word orders in Turkish. In *Proceedings of the Seventh International Generation Workshop*.

Xiarong Huang. 1994. Planning reference choices for argumentative texts. In *Seventh International Workshop on Natural Language Generation*, pages 145–152, June.

Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. 1991. Lexical selection and paraphrase in a meaning-text generation model. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 293–312. Kluwer, Dordrecht.

Ray S. Jackendoff. 1990. *Semantic structures*. MIT Press, Cambridge, MA.

Aravind K. Joshi, L. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10:136–163.

Aravind K. Joshi. 1987. The relevance of tree adjoining grammar to generation. In Gerard Kempen, editor, *Natural Language Generation*, pages 233–252. Martinus Nijhoff Press, Dordrect, The Netherlands.

Richard Kittredge, Tanya Korelsky, and Owen Rambow. 1991. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314.

Amichai Kronfeld. 1986. Donellan's distinction and a computational model of reference. In *Proceedings of ACL*, pages 186–191.

Christian M. I. M. Mathiessen. 1983. Systemic grammar in computation: the Nigel case. In *Proceedings of EACL*, pages 155–164.

David D. McDonald and James D. Pustejovsky. 1985. TAG's as a grammatical formalism for generation. In *Proceedings of the 23rd Annual Meeting of the*

Association for Computational Linguistics, pages 94–103, Chicago, IL.

David McDonald. 1992. Type-driven suppression of redundancy in the generation of inference-rich reports. In Robert Dale, Eduard Hovy, Dietmar Rösner, and Oliviero Stock, editors, *Aspects of Automated Natural Language Generation: 6th International Workshop on Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 73–88. Springer Verlag, Berlin.

Igor A. Mel'čuk and Alain Polguère. 1987. A formal lexicon in the meaning-text theory (or how to do lexica with words). *Computational Linguistics*, 13(3–4):261–275.

Nicolas Nicolov, Chris Mellish, and Graeme Ritchie. 1995. Sentence generation from conceptual graphs. In W. Rich G. Ellis, R. Levinson and F. Sowa, editors, *Conceptual Structures: Applications, Implementation and Theory (Proceedings of Third International Conference on Conceptual Structures)*, pages 74–88. Springer.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

Barbara H. Partee. 1973. Some structural analogies between tenses and pronouns in English. *Journal of Philosophy*, 70:601–609.

Barbara H. Partee. 1984. Nominal and temporal anaphora. *Linguistics and Philosophy*, 7(3):243–286.

Scott Prevost and Mark Steedman. 1993. Generating contextually appropriate intonation. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*.

Ellen Prince. 1981. Toward a taxonomy of given-new information. In P. Cole, editor, *Radical Pragmatics*. Academic Press.

Ellen Prince. 1986. On the syntactic marking of presupposed open propositions. In *Proceedings of the 22nd Annual Meeting of the Chicago Linguistic Society*, pages 208–222, Chicago. CLS.

Ellen Prince. 1993. On the functions of left dislocation. Manuscript, University of Pennsylvania.

James Pustejovsky. 1991. The generative lexicon. *Computational Linguistics*, 17(3):409–441.

Ehud Reiter and Robert Dale. 1992. A fast algorithm for the generation of referring expressions. In *Proceedings of COLING*, pages 232–238.

Ehud Reiter. 1991. A new model of lexical choice for nouns. *Computational Intelligence*, 7(4):240–251.

Craige Roberts. 1986. *Modal Subordination, Anaphora and Distributivity*. Ph.D. thesis, University of Massachusetts, Amherst.

Eleanor Rosch. 1978. Principles of categorization. In Eleanor Rosch and Barbara B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Erlbaum, Hillsdale, NJ.

Robert Rubinoff. 1992. Integrating text planning and linguistic choice by annotating linguistic structures. In Robert Dale, Eduard Hovy, Dietmar Rösner, and Oliviero Stock, editors, *Aspects of Automated Natural Language Generation: 6th International Workshop on Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 45–56. Springer Verlag, Berlin.

Yves Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, Computer Science Department, University of Pennsylvania.

Stuart Shieber and Yves Schabes. 1991. Generation and synchronous tree adjoining grammars. *Computational Intelligence*, 4(7):220–228.

Stuart Shieber, Gertjan van Noord, Fernando Pereira, and Robert Moore. 1990. Semantic-head-driven generation. *Computational Linguistics*, 16:30–42.

Frank Smadja and Kathleen McKeown. 1991. Using collocations for language generation. *Computational Intelligence*, 7(4):229–239.

Evelyne Viegas and Pierrette Bouillon. 1994. Semantic lexicons: the cornerstone for lexical choice in natural language generation. In *Seventh International Workshop on Natural Language Generation*, pages 91–98, June.

K. Vijay-Shanker. 1987. *A Study of Tree Adjoining Grammars*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Wolfgang Wahlster, Elisabeth André, Son Bandyopadhyay, Winfried Graf, and Thomas Rist. 1991. WIP: The coordinated generation of multimodal presentations from a common representation. In Oliviero Stock, John Slack, and Andrew Ortony, editors, *Computational Theories of Communication and their Applications*. Berlin: Springer Verlag.

Leo Wanner. 1994. Building another bridge over the generation gap. In *Seventh International Workshop on Natural Language Generation*, pages 137–144, June.

Gregory Ward and Ellen Prince. 1991. On the topicalization of indefinite NPs. *Journal of Pragmatics*, 15(8):338–351.

Gregory Ward. 1985. *The Semantics and Pragmatics of Preposing*. Ph.D. thesis, University of Pennsylvania. Published 1988 by Garland.

Janyce M. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Gijoo Yang, Kathleen F. McCoy, and K. Vijay-Shanker. 1991. From functional specification to syntactic structures: systemic grammar and tree-adjoining grammar. *Computational Intelligence*, 7(4):207–219.